CLAIMS

We claim:

1.   In a hierarchical classification system performing caching, a method comprising:

walking a classification tree in a hierarchical classification system to determine whether an incoming flow matches a class in the classification tree; and

performing a lookup on a cache storing a data structure of a plurality of classes of one classification type to compare the incoming flow with a plurality of classes at the same time to determine whether the incoming flow matches one of the plurality of classes.

2.   The method defined in Claim 1 wherein the data structure comprises a hash table.

3.   The method defined in Claim 1 further comprising returning a class pointer indicative of user programming information that has been assigned to a class if the incoming flow matches the class.

4.   The method defined in Claim 1 performing a walk through the plurality of classes if a determination that the class in the plurality of classes is

not known as a result of performing the cache lookup.

5. The method defined in Claim 4 wherein if the class of the incoming packet hits the cache, then further comprising:

storing the class in the cache; and

returning a result indicating the class was in the cache.

6. The method defined in Claim 4 wherein if the class of the incoming packet does not hit the cache, further comprising:

marking the cache result as indicating that the class is not in the cache; and

continuing to walk the classification tree from a location in the tree immediately after the end of the portion of the tree represented in the cache.

7. An apparatus for use in a hierarchical classification system performing caching, the apparatus comprising:

a memory storing a classification tree;

a cache to store a cacheable portion of the classification tree; and

a classification engine coupled to the memory to walk the classification tree as part of a hierarchical classification system to determine whether an incoming flow matches a class in the classification tree, and to perform a lookup on the cache storing a data structure of a plurality of classes of one

classification type to compare the incoming flow with a plurality of classes at the same time to determine whether the incoming flow matches one of the plurality of classes.

8.   The apparatus defined in Claim 7 wherein the data structure comprises a hash table.

9.   The apparatus defined in Claim 7 wherein the classification engine returns a class pointer indicative of user programming information that has been assigned to a class if the incoming flow matches the class.

10.   The apparatus defined in Claim 7 wherein the classification engine performs a walk through the plurality of classes if a determination that the class in the plurality of classes is not known as a result of performing the cache lookup.

11.   The apparatus defined in Claim 10 wherein if the class of the incoming packet hits the cache, then the classification engine:

stores the class in the cache; and

returns a result indicating the class was in the cache.

12.   The apparatus defined in Claim 10 wherein if the class of the

incoming packet does not hit the cache, then the classification engine:

marks the cache result as indicating that the class is not in the cache; and

continues to walk the classification tree from a location in the tree

immediately after the end of the portion of the tree represented in the cache.

13.   An apparatus for use in a hierarchical classification system

performing caching, the apparatus comprising:

means for walking a classification tree in a hierarchical classification

system to determine whether an incoming flow matches a class in the

classification tree; and

means for performing a lookup on a cache storing a data structure of a

plurality of classes of one classification type to compare the incoming flow

with a plurality of classes at the same time to determine whether the

incoming flow matches one of the plurality of classes.

14.   An article of manufacture having one or more recordable media

with executable instructions stored thereon which, when executed by a

system, cause the system to:

walk a classification tree in a hierarchical classification system to

determine whether an incoming flow matches a class in the classification tree;

and

perform a lookup on a cache storing a data structure of a plurality of classes of one classification type to compare the incoming flow with a plurality of classes at the same time to determine whether the incoming flow matches one of the plurality of classes.

15.   In a hierarchical classification system performing caching, a method for classifying a flow comprising:

ordering a classification tree of classification types by grouping at least a portion of the classes of each type of classification together in the classification tree;

creating a data structure of a plurality of classes of one classification type to determine whether the flow matches any items in the data structure;

performing tree-walking of the classification tree; and

performing a lookup on a cache storing the data structure to compare the incoming flow with a plurality of classes at the same time to determine whether the incoming flow matches one of the plurality of classes.

16.   The method defined in Claim 15 wherein the data structure comprises a hash table.

17.   The method defined in Claim 15 wherein the classification types in the classification tree includes addresses.

18. The method defined in Claim 17 wherein the classification types in the classification tree includes one or more services.

19. The method defined in Claim 18 wherein the classification types in the classification tree further include ports.

20. The method defined in Claim 17 wherein the addresses comprise Internet Protocol (IP) addresses.

21. The method defined in Claim 15 wherein the classification types in the classification tree includes services and ports.